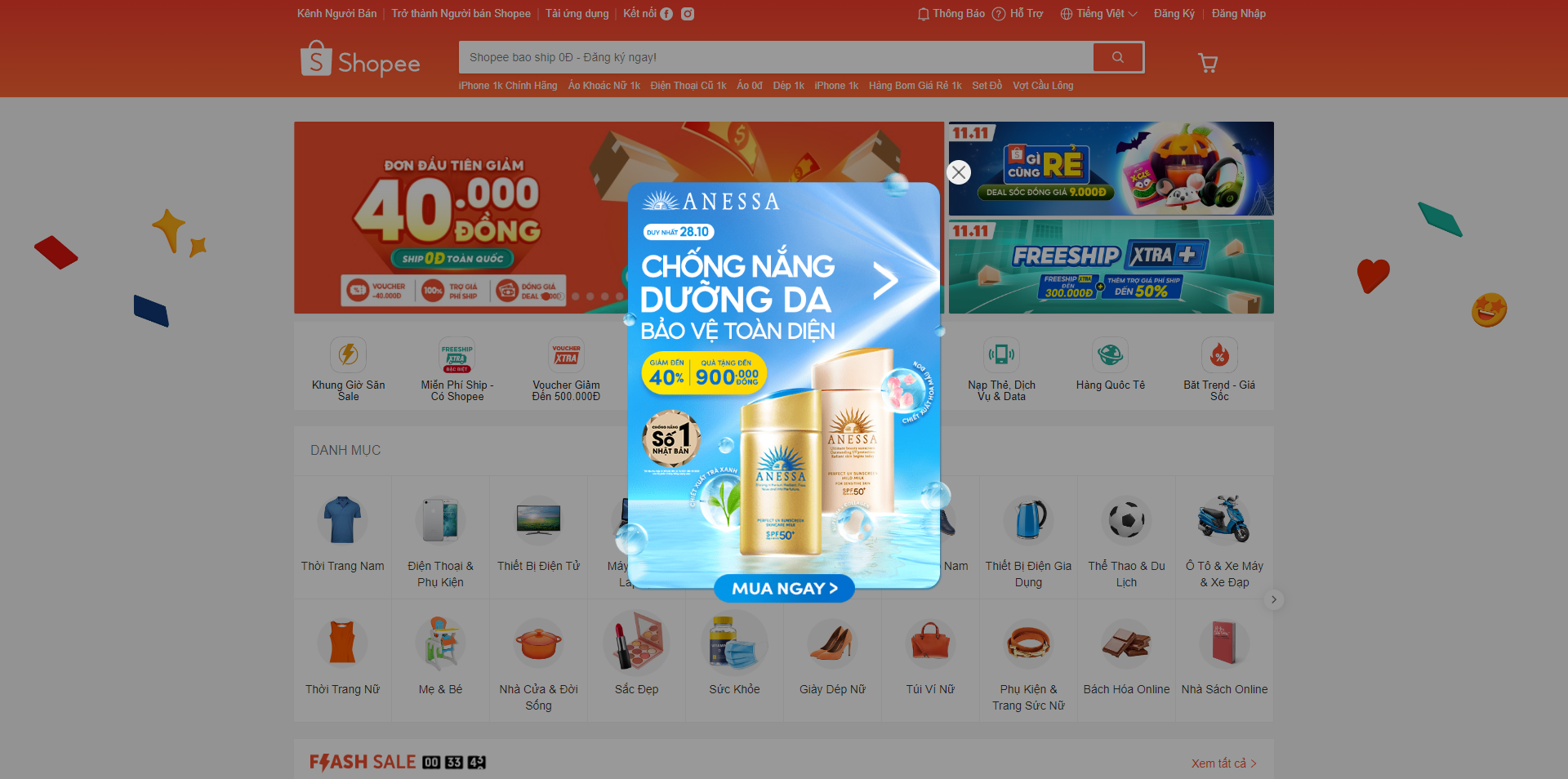
**CHUYÊN NGÀNH LẬP TRÌNH WEB (LÀ MỘT NGHỀ HOT):**

Tạo một ứng dụng web hiện đại thường được chia thành ba phần quan trọng: phần Front-End (FE), phần Back-End (BE), và Cơ sở dữ liệu (Database). Dưới đây, tôi sẽ giải thích từng phần:

**1.Front-End (FE):**

· Mô tả: Front-End là phần của ứng dụng web mà người dùng cuối thấy và tương tác. Nó bao gồm giao diện người dùng (UI) và trải nghiệm người dùng (UX).



**ĐÂY GỌI LÀ GIAO DIỆN WEB, NHỮNG HIỆU ỨNG, FONT CHỮ, CÁC HÌNH TRÒN, HÌNH VUÔNG,...CHÍNH LÀ DO FRONT END ĐẢM NHIỆM**

· Một ví dụ mô tả về hoạt động của Front-End (FE) trong một ứng dụng web:

**Ví dụ:** Hiển thị danh sách sản phẩm trên trang mua sắm trực tuyến

1. **Người dùng truy cập trang mua sắm:** Người dùng truy cập trang mua sắm trực tuyến thông qua trình duyệt web. Họ muốn xem danh sách các sản phẩm để mua.

2. **Front-End gửi yêu cầu cho Back-End**: Front-End gửi yêu cầu cho Back-End để lấy danh sách các sản phẩm. Yêu cầu này có thể được gửi thông qua một API hoặc URL cụ thể.

3. **Back-End xử lý yêu cầu:** Back-End (BE) nhận yêu cầu từ FE, sau đó truy vấn cơ sở dữ liệu để lấy thông tin về các sản phẩm. Sau khi BE đã lấy dữ liệu từ cơ sở dữ liệu, nó trả lại danh sách sản phẩm cho FE.

4. **Front-End hiển thị danh sách sản phẩm:** FE nhận dữ liệu sản phẩm từ BE và sử dụng ngôn ngữ lập trình như JavaScript và thư viện hoặc framework như React, Angular, hoặc Vue.js để hiển thị danh sách sản phẩm trên trang web. FE xử lý việc định dạng và hiển thị dữ liệu sao cho nó dễ đọc và trực quan cho người dùng.

5. **Người dùng xem danh sách sản phẩm:** Sau khi FE đã hiển thị danh sách sản phẩm, người dùng có thể xem thông tin về các sản phẩm, bao gồm tên, hình ảnh, giá cả và mô tả. Họ có thể thực hiện các tương tác như chọn sản phẩm, xem chi tiết và thêm vào giỏ hàng.

Trong ví dụ này, Front-End đảm nhiệm vai trò quan trọng trong hiển thị thông tin sản phẩm cho người dùng cuối và tạo trải nghiệm người dùng trên trình duyệt web. FE sử dụng ngôn ngữ lập trình và công cụ để tạo giao diện người dùng trực quan và tương tác cho người dùng.

· Ngôn ngữ và Công Cụ Chính:

* HTML (Hypertext Markup Language): Định dạng và hiển thị nội dung trên trình duyệt.
* CSS (Cascading Style Sheets): Định dạng và thiết kế giao diện người dùng, làm cho trang web trở nên hấp dẫn hơn.
* JavaScript: Thêm tính năng tương tác và động cho trang web.

· Frameworks và Thư viện:

* React, Angular, và Vue.js: Các frameworks phổ biến cho việc xây dựng ứng dụng web đơn trang (Single-Page Applications - SPAs). (Sau này, xin việc làm sẽ phụ thuộc vào bạn đang làm về framework nào.)
* jQuery: Thư viện JavaScript giúp quản lý DOM và thao tác với nội dung trên trang web dễ dàng hơn.

· Figma:

* Nhiệm vụ: Figma là một công cụ thiết kế và làm việc cùng đồng đội (collaborative design tool). Nó được sử dụng để thiết kế giao diện người dùng (UI) của ứng dụng web và tạo wireframe, mockup, và prototype cho các trang web và ứng dụng.
* Vai trò: Figma giúp người thiết kế và phát triển đội làm việc cùng nhau để thiết kế và đánh giá giao diện người dùng trước khi bắt đầu phát triển. Nó tạo điều kiện cho việc hiểu rõ cách ứng dụng web sẽ trông như và hoạt động trước khi code được viết.

· Responsive Design: Đảm bảo trang web hoạt động trên nhiều thiết bị và kích thước màn hình khác nhau (desktop, điện thoại di động, máy tính bảng).

· **Ví dụ thực tế: Trang web thương mại điện tử như Amazon sử dụng HTML, CSS và JavaScript để hiển thị sản phẩm, cho phép người dùng tìm kiếm và thêm sản phẩm vào giỏ hàng.**

**2. Back-End (BE):**

· **Mô tả:** Back-End là phần của ứng dụng web nằm phía sau và xử lý dữ liệu, tương tác với cơ sở dữ liệu, và cung cấp dữ liệu cho Front-End.

· **Ví dụ** một hoạt động quan trọng của Back-End (BE) trong phát triển ứng dụng web là xử lý và quản lý dữ liệu.

**Ví dụ: Đăng nhập vào ứng dụng web**

1. **Người dùng truy cập trang đăng nhập:** Người dùng truy cập trang đăng nhập trên Front-End (FE) của ứng dụng web để nhập thông tin đăng nhập, chẳng hạn như tên người dùng và mật khẩu.

2. **Front-End gửi yêu cầu đến Back-End:** Sau khi người dùng nhập thông tin đăng nhập và nhấn nút "Đăng nhập," FE gửi yêu cầu đăng nhập đến Back-End thông qua một API hoặc URL cụ thể.

3. **Back-End kiểm tra thông tin đăng nhập:** Back-End nhận yêu cầu, kiểm tra thông tin đăng nhập với cơ sở dữ liệu. Điều này bao gồm truy vấn cơ sở dữ liệu để kiểm tra xem tên người dùng và mật khẩu có khớp với dữ liệu trong cơ sở dữ liệu hay không.

4. **Xác thực và phản hồi:** Nếu thông tin đăng nhập hợp lệ, BE sẽ xác thực người dùng và tạo phiên làm việc (session) cho họ. Sau đó, BE sẽ gửi phản hồi (response) cho FE để cho biết đăng nhập thành công.

5. **Truy cập ứng dụng:** Sau khi FE nhận được phản hồi từ BE cho biết đăng nhập thành công, người dùng sẽ được chuyển đến các trang có quyền truy cập, và họ có thể bắt đầu sử dụng ứng dụng.

Trong ví dụ này, Back-End đảm nhận vai trò quan trọng trong xử lý thông tin đăng nhập của người dùng, kiểm tra tính hợp lệ và tương tác với cơ sở dữ liệu để xác thực người dùng. Nó là một ví dụ cơ bản về hoạt động của Back-End trong quá trình xử lý và quản lý dữ liệu trong ứng dụng web.

· Ngôn ngữ và Công Cụ Chính:

Ngôn ngữ lập trình thường được sử dụng cho phần Back-End.

* JavaScript (Node.js): Phát triển phần mềm máy chủ.
* Python (Django, Flask): Xây dựng ứng dụng web và API.
* Ruby (Ruby on Rails): Phát triển ứng dụng web nhanh chóng.
* PHP() Lavarel, Symfony : Sử dụng trong một số ứng dụng web truyền thống.
* Java(Spring Java): Nó được sử dụng rộng rãi trong phát triển các ứng dụng web, ứng dụng di động và các dịch vụ web.

**\*NOTE:** C# (ASP.NET): Sử dụng phổ biến trong phát triển ứng dụng web trên nền tảng Microsoft.

· APIs (Application Programming Interfaces): Back-End cung cấp APIs để truy xuất và gửi dữ liệu giữa Front-End và Back-End

**3. Cơ Sở Dữ Liệu (Database):**

· Mô tả: Cơ sở dữ liệu lưu trữ dữ liệu của ứng dụng web, bao gồm thông tin người dùng, nội dung, và dữ liệu khác.

· Hệ quản trị cơ sở dữ liệu (DBMS): Các hệ quản trị cơ sở dữ liệu phổ biến bao gồm MySQL, PostgreSQL, MongoDB (NoSQL), Microsoft SQL Server, và nhiều lựa chọn khác.

· Ngôn ngữ truy vấn cơ sở dữ liệu: Sử dụng SQL (Structured Query Language) để truy vấn và tương tác với cơ sở dữ liệu.

· ORMs (Object-Relational Mapping): Sử dụng các ORM như Hibernate (cho Java), Sequelize (cho Node.js), và Entity Framework (cho .NET) để tương tác với cơ sở dữ liệu theo cách dễ dàng và hướng đối tượng. - Optimization: Tối ưu hóa cơ sở dữ liệu để cải thiện hiệu suất và đảm bảo dữ liệu được lưu trữ một cách hiệu quả.

· Một ví dụ mô tả về hoạt động của cơ sở dữ liệu trong một ứng dụng web:

**Ví dụ: Lưu trữ và quản lý thông tin người dùng trong ứng dụng xã hội**

1. Người dùng đăng ký tài khoản: Một người dùng mới truy cập trang đăng ký tài khoản của ứng dụng xã hội. Họ cung cấp thông tin cá nhân như tên, địa chỉ email và mật khẩu.

2. Lưu trữ thông tin người dùng: Sau khi người dùng hoàn thành đăng ký, các thông tin cá nhân này cần được lưu trữ. Dữ liệu này bao gồm thông tin về tài khoản người dùng, bao gồm tên, địa chỉ email và mật khẩu.

3. Gửi yêu cầu lưu trữ đến cơ sở dữ liệu: Front-End (FE) của ứng dụng gửi yêu cầu lưu trữ thông tin người dùng đến Back-End (BE) của ứng dụng, thông qua một API hoặc giao thức cụ thể.

4. Cơ sở dữ liệu lưu trữ thông tin: Cơ sở dữ liệu (DB) nhận yêu cầu từ BE để lưu trữ thông tin người dùng. DB sẽ tạo một bản ghi mới trong bảng dữ liệu "Người dùng" và lưu thông tin tài khoản người dùng bao gồm tên, địa chỉ email và mật khẩu. Dữ liệu này được lưu trữ an toàn trong DB.

5. Xác thực và truy xuất dữ liệu: Khi người dùng đăng nhập sau khi đã đăng ký, FE gửi yêu cầu xác thực đăng nhập cho BE, và BE sẽ kiểm tra thông tin đăng nhập với dữ liệu từ cơ sở dữ liệu. Nếu thông tin đúng, người dùng được xác thực và có quyền truy cập vào tài khoản của họ.

6. Truy cập và cập nhật thông tin: Khi người dùng đăng nhập, FE gửi yêu cầu truy cập thông tin tài khoản, và BE sẽ truy cập DB để lấy thông tin cá nhân của người dùng. Người dùng có thể cập nhật thông tin tài khoản của họ, chẳng hạn như thay đổi địa chỉ email hoặc mật khẩu, và thông tin này cũng sẽ được cập nhật trong cơ sở dữ liệu.

Trong ví dụ này, cơ sở dữ liệu (DB) đóng vai trò quan trọng trong việc lưu trữ và quản lý thông tin người dùng, cung cấp dữ liệu cho việc xác thực và truy cập, và đảm bảo tính an toàn của dữ liệu người dùng trong ứng dụng xã hội.

**\*NOTE:**

· Nhớ rằng, trong phát triển ứng dụng web thực tế, Front-End, Back-End và Cơ sở dữ liệu thường hoạt động cùng nhau để cung cấp trải nghiệm người dùng tốt nhất và quản lý dữ liệu hiệu quả.

· **Trong doanh nghiệp thường sẽ có 3 bộ phận:** Front-End (FE), phần Back-End (BE), và Cơ sở dữ liệu (Database).